Estimating elliptic flow coefficient in heavy ion collisions using deep learning

ALICE Physics Week
25-29 July 2022

Neelkamal Mallick
Indian Institute of Technology Indore, India
neelkamal.mallick@cern.ch

Based on:

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Outline

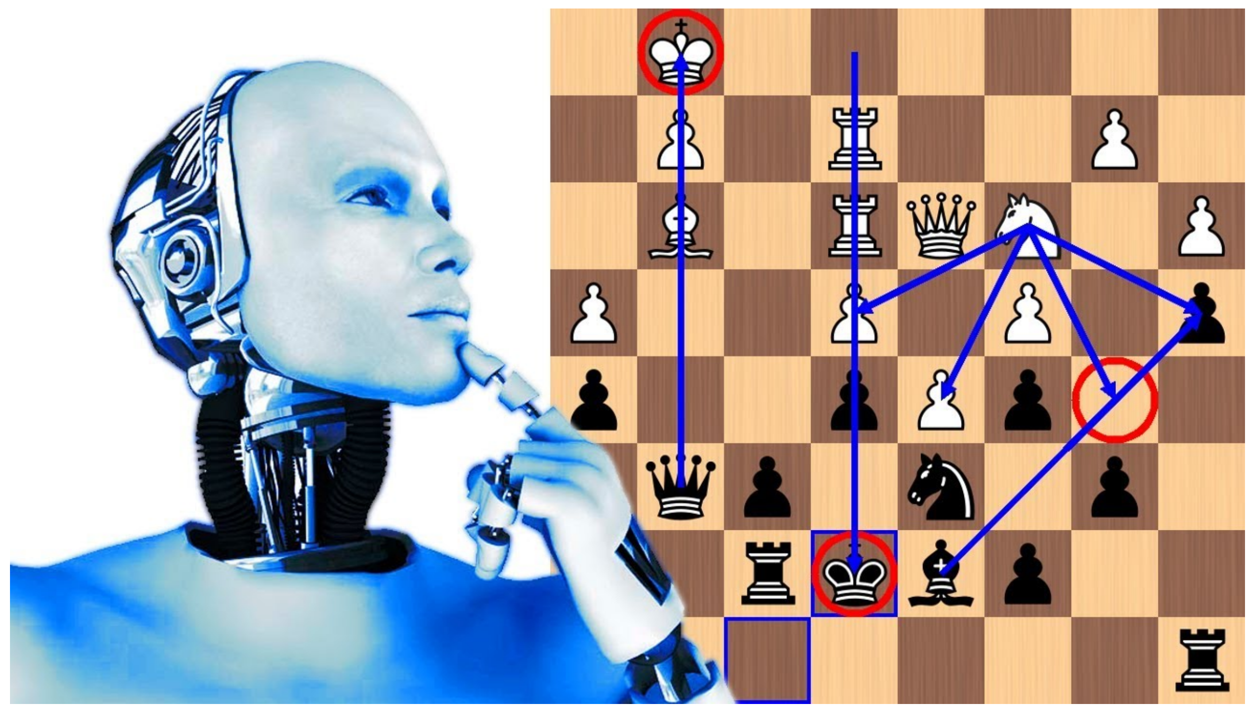
- Introduction
- Motivation
- Deep Learning Estimator
- Results
- Summary

Introduction

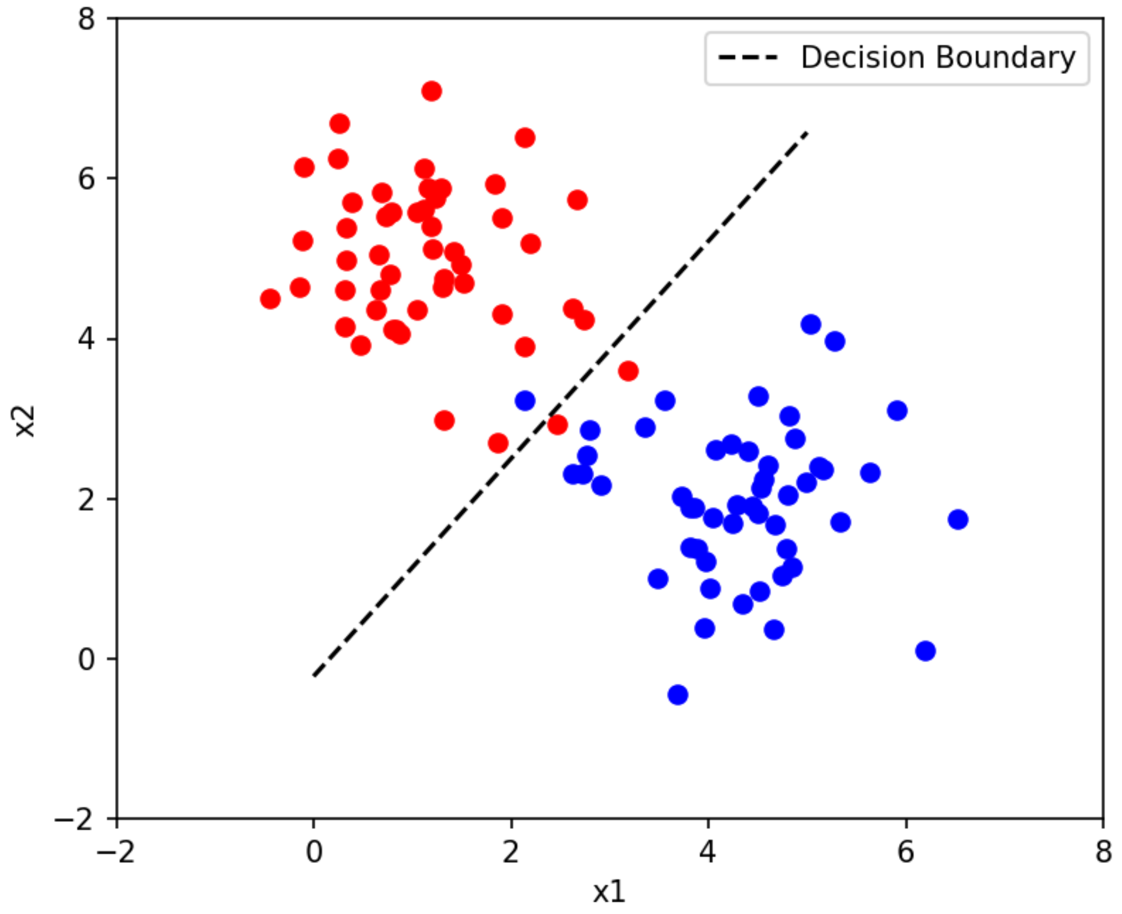
"Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed."

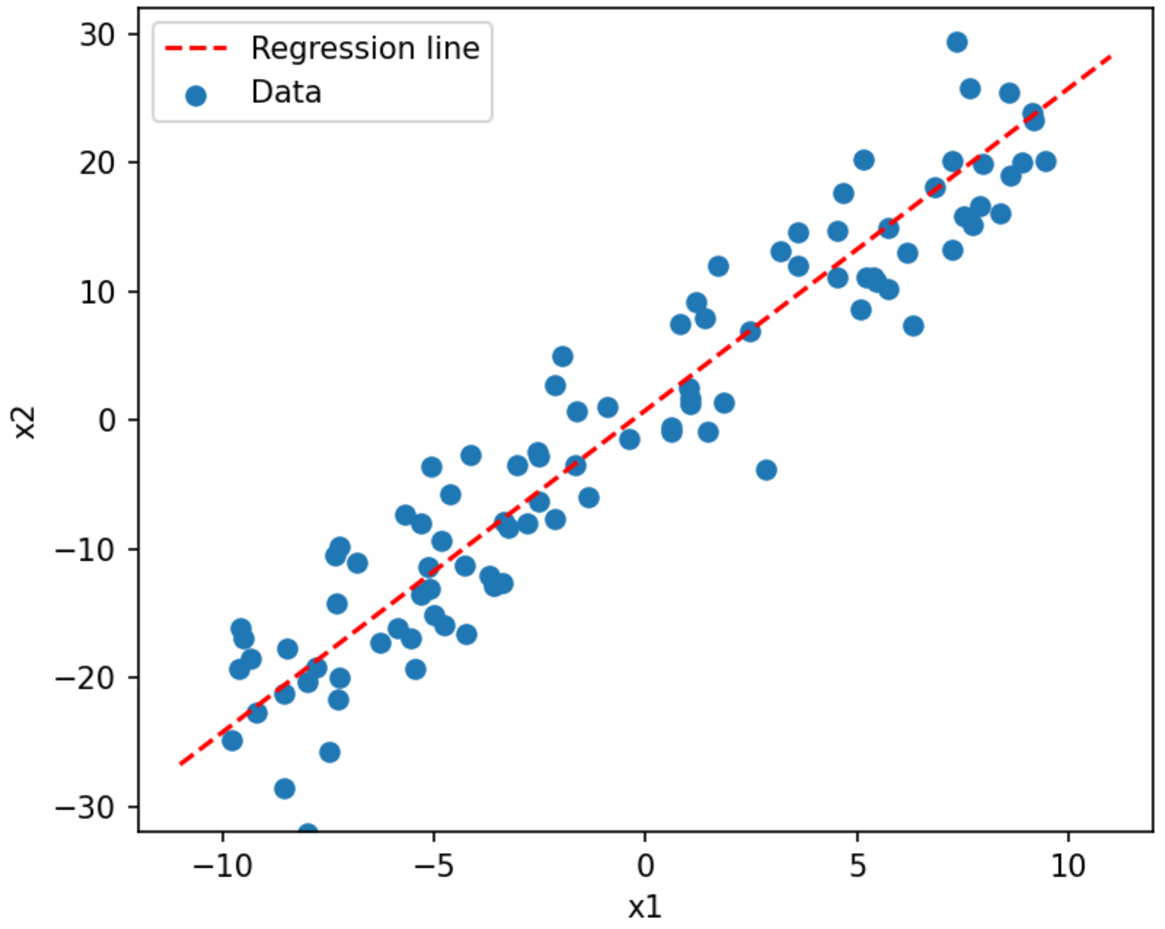
-Arthur Samuel, 1959

- Big data
- Learning algorithm (BDT, DNN, *etc.*)
- Knowledge from data
- Tune the parameters (Optimise the model)
- Predict!!



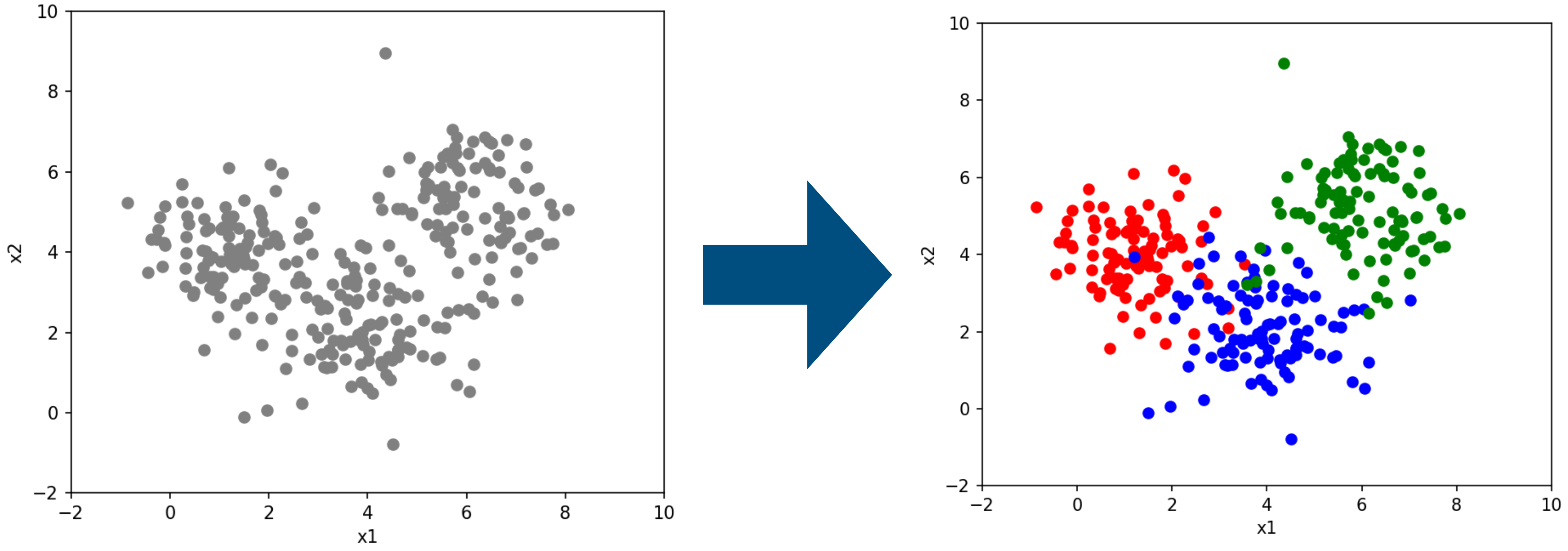
Cat or Dog?



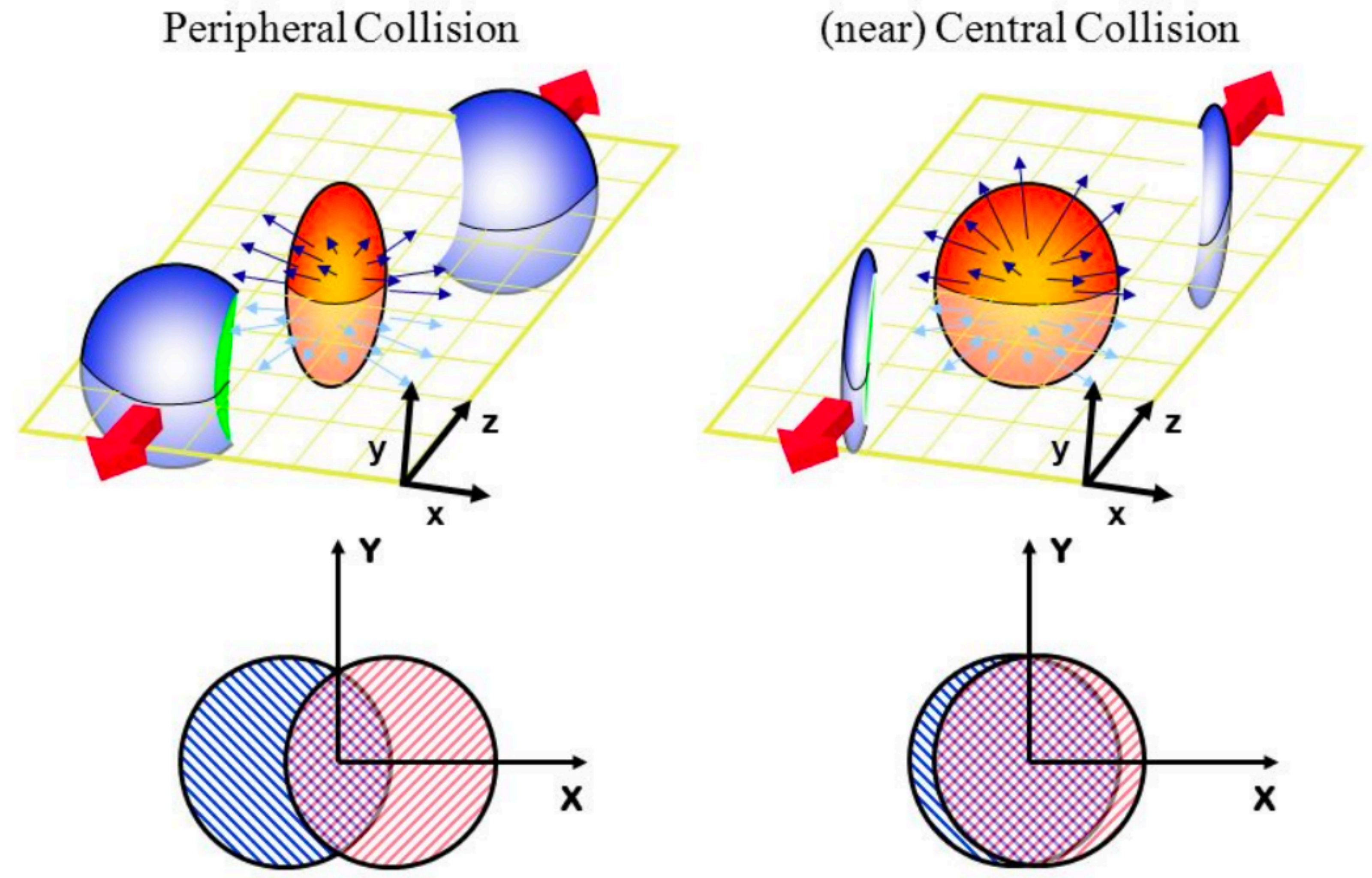


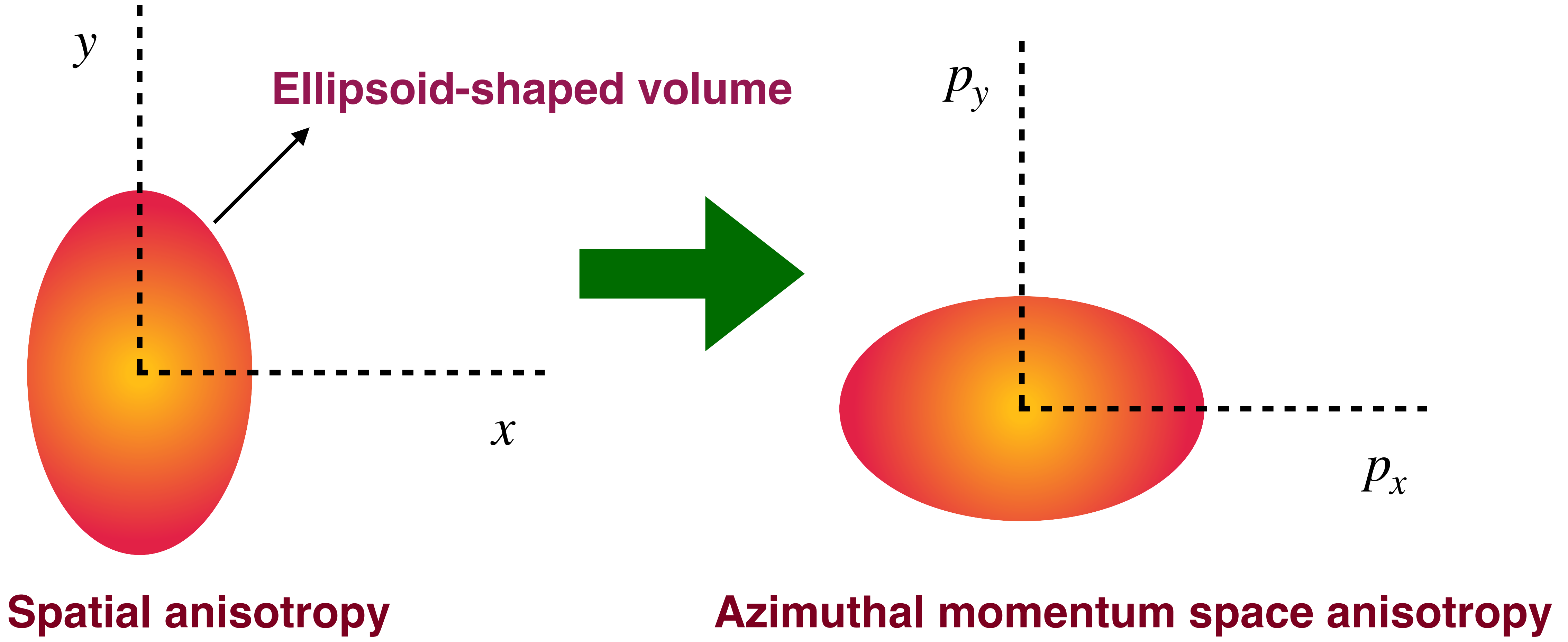
Supervised/unsupervised

- Classification
- Regression
- Clustering *etc.*



Elliptic Flow

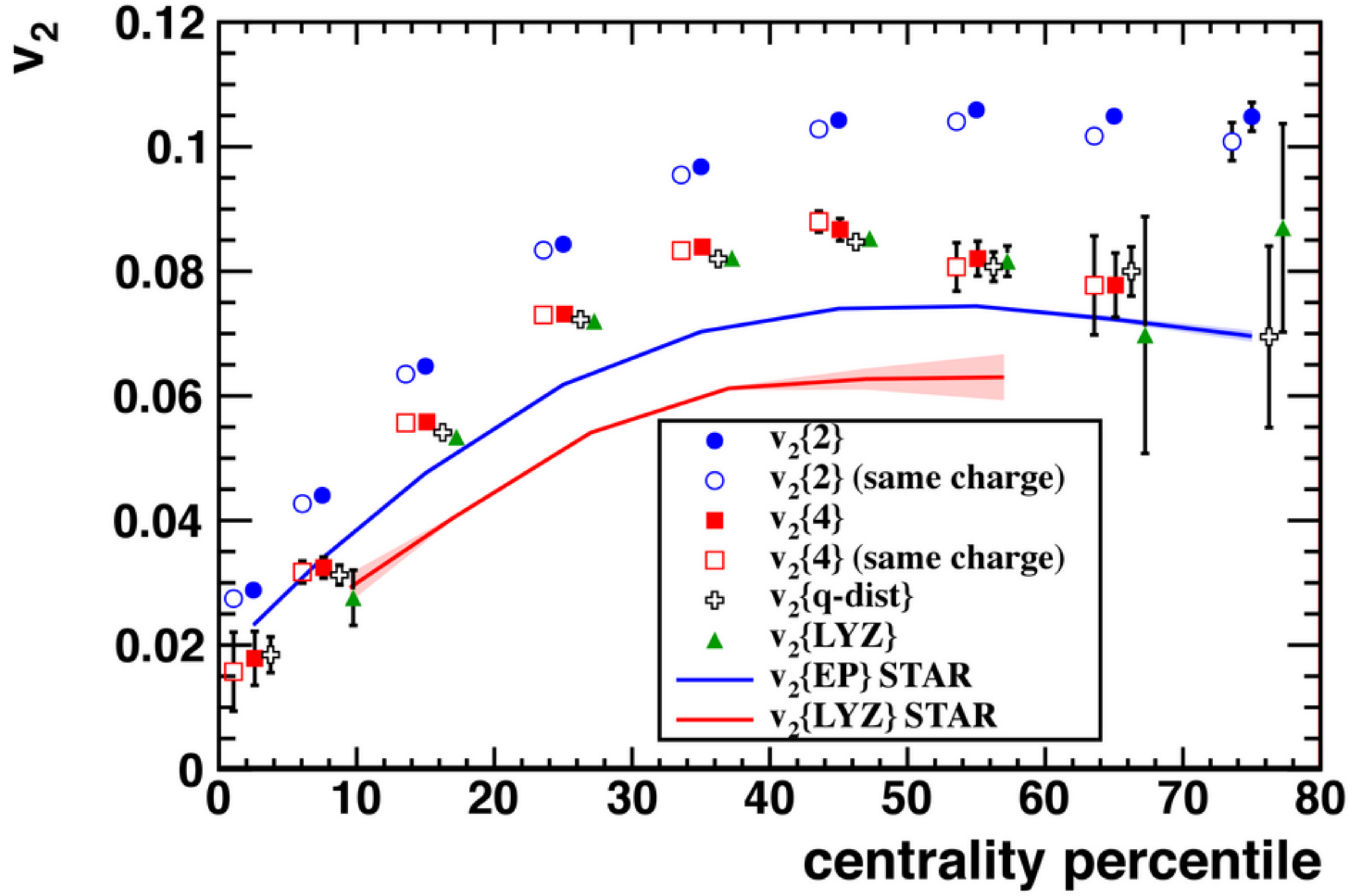




- Initial state has ***spatial anisotropy*** in the transverse (xy) plane
- ***Spatial anisotropy*** creates ***anisotropic pressure gradients***
- Final state characterised by ***momentum anisotropy*** ***(v_1, v_2, v_3… etc.)***

$$E\frac{d^3N}{dp^3} = \frac{d^3N}{p_\mathrm{T}dp_\mathrm{T}dyd\phi} = \frac{d^2N}{p_\mathrm{T}dp_\mathrm{T}dy}\frac{1}{2\pi}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos[n(\phi - \psi_n)]\right)$$

$$v_n(p_\mathrm{T}, y) = \langle \cos(n(\phi - \psi_n)) \rangle$$

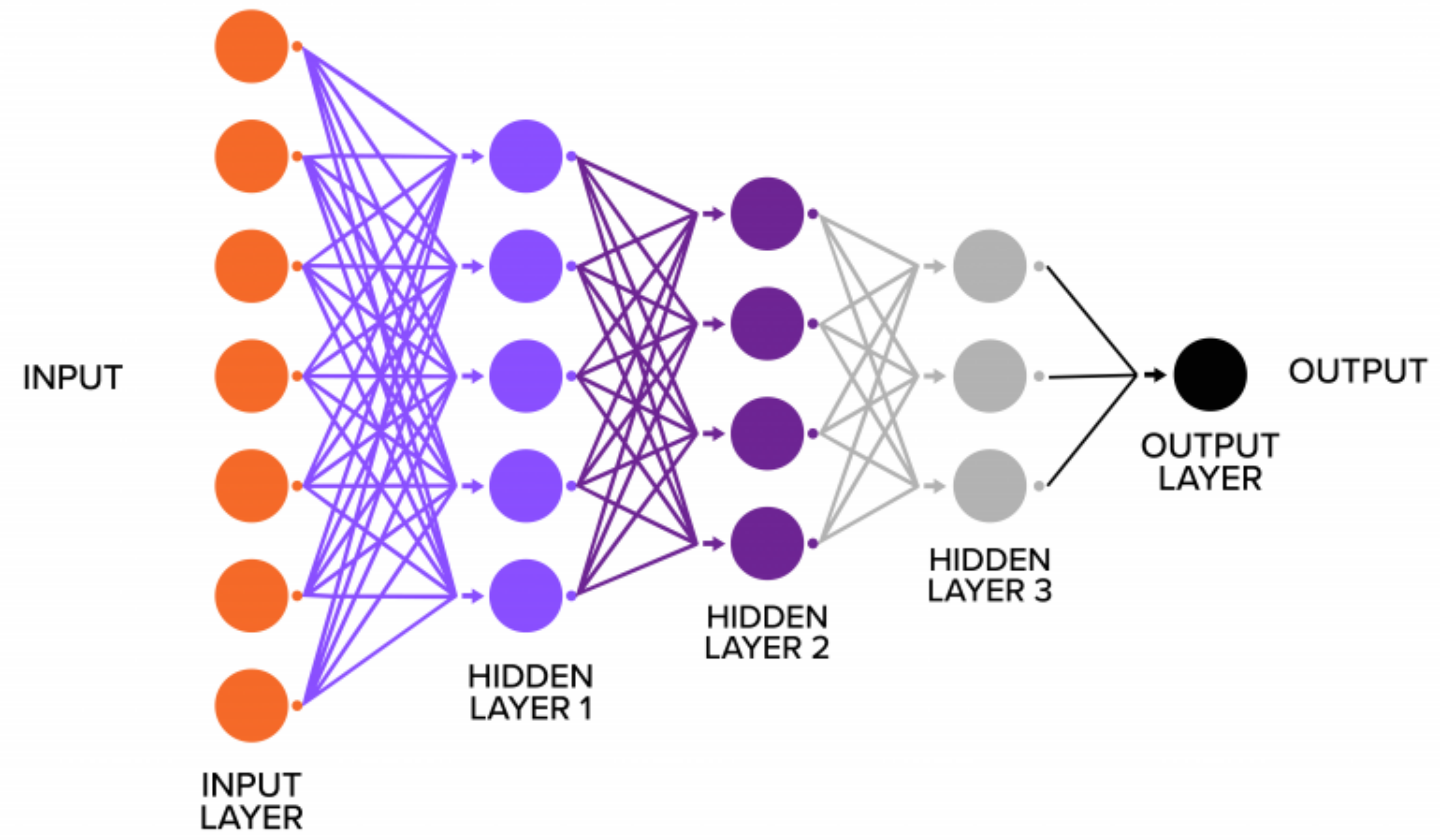


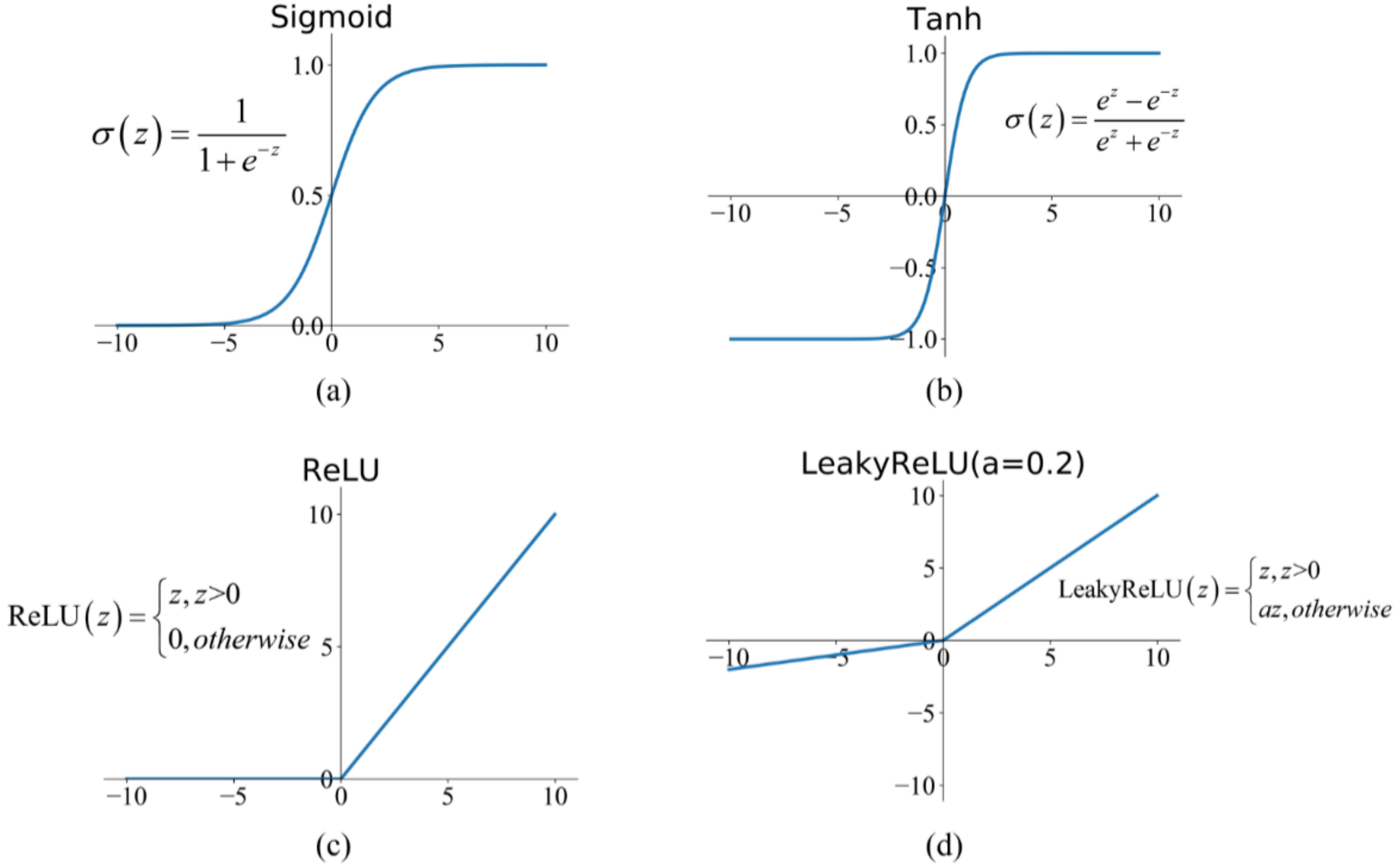
ALICE, Phys. Rev. Lett. 105, 252302 (2010).

Deep Neural Network

- Learn the mapping function, $y = \mathbf{f}^*(\mathbf{x})$ or $\mathbf{y} = \mathbf{f}(\mathbf{x}; \theta)$
- Three key layers
  Input: Takes the features as input
  Hidden layers: Connects to each node through different weights
  Output: Gives the result as a number or class
- **Weights** dictate the importance of an input → more important features get more weights
- **Activation function**: Includes nonlinearity in the model
- **Cost function**: Evaluates the accuracy between machine prediction and true value
- **Optimizer**: Method (or algorithm) that minimizes the cost function by automatically updating the weights

$$\mathbf{y} = f(\langle \mathbf{x}, \mathbf{W} \rangle + b) \qquad \mathbf{y} = \mathbf{f}^{(3)}(\mathbf{f}^{(2)}(\mathbf{f}^{(1)}(\,.\,)))$$

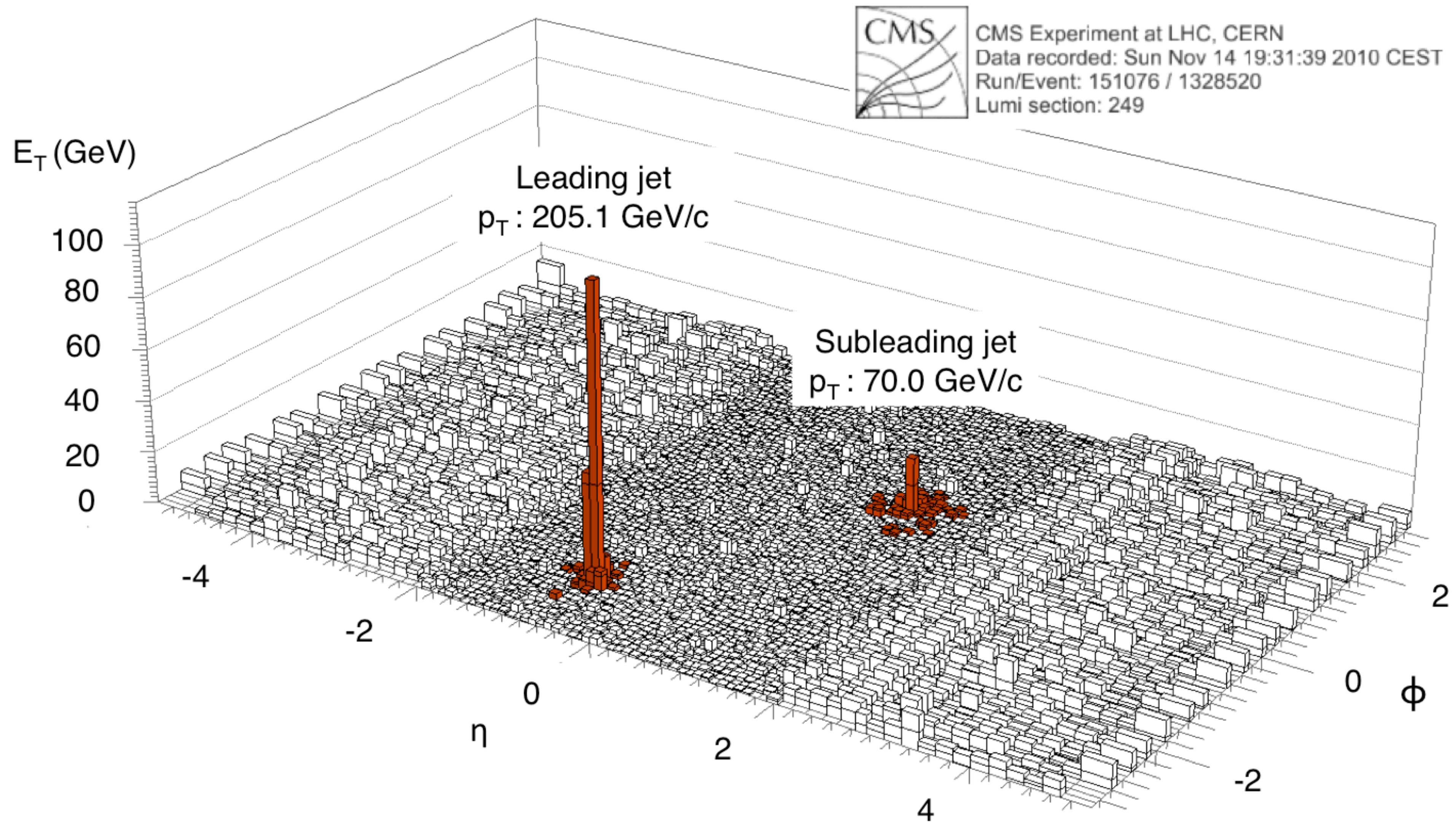




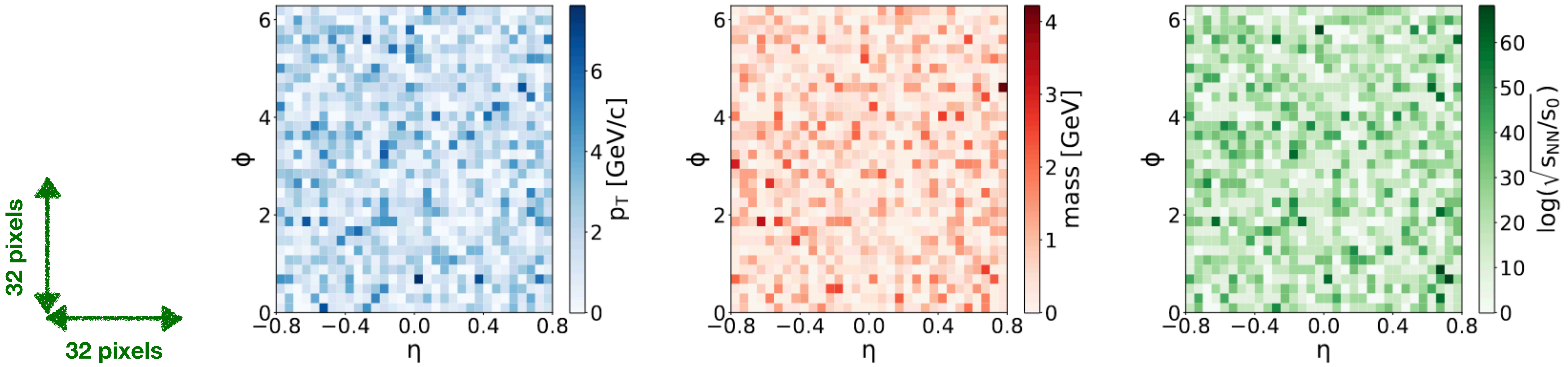
Junxi Feng et al. , Phys. Rev. E 100, 033308 (2019).

Deep Learning Estimator

- First Deep Neural Network based estimator for v_2
- $(\eta - \phi)$ space as the primary input space
- p_{T}, mass, and $\log(\sqrt{s_{\mathrm{NN}}/s_0})$ weighted layers serve as the secondary input space
- Model trained on Pb-Pb, $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV (Min. Bias)



Serguei Chatrchyan et al., Phys.Rev.C 84, 024906 (2011)

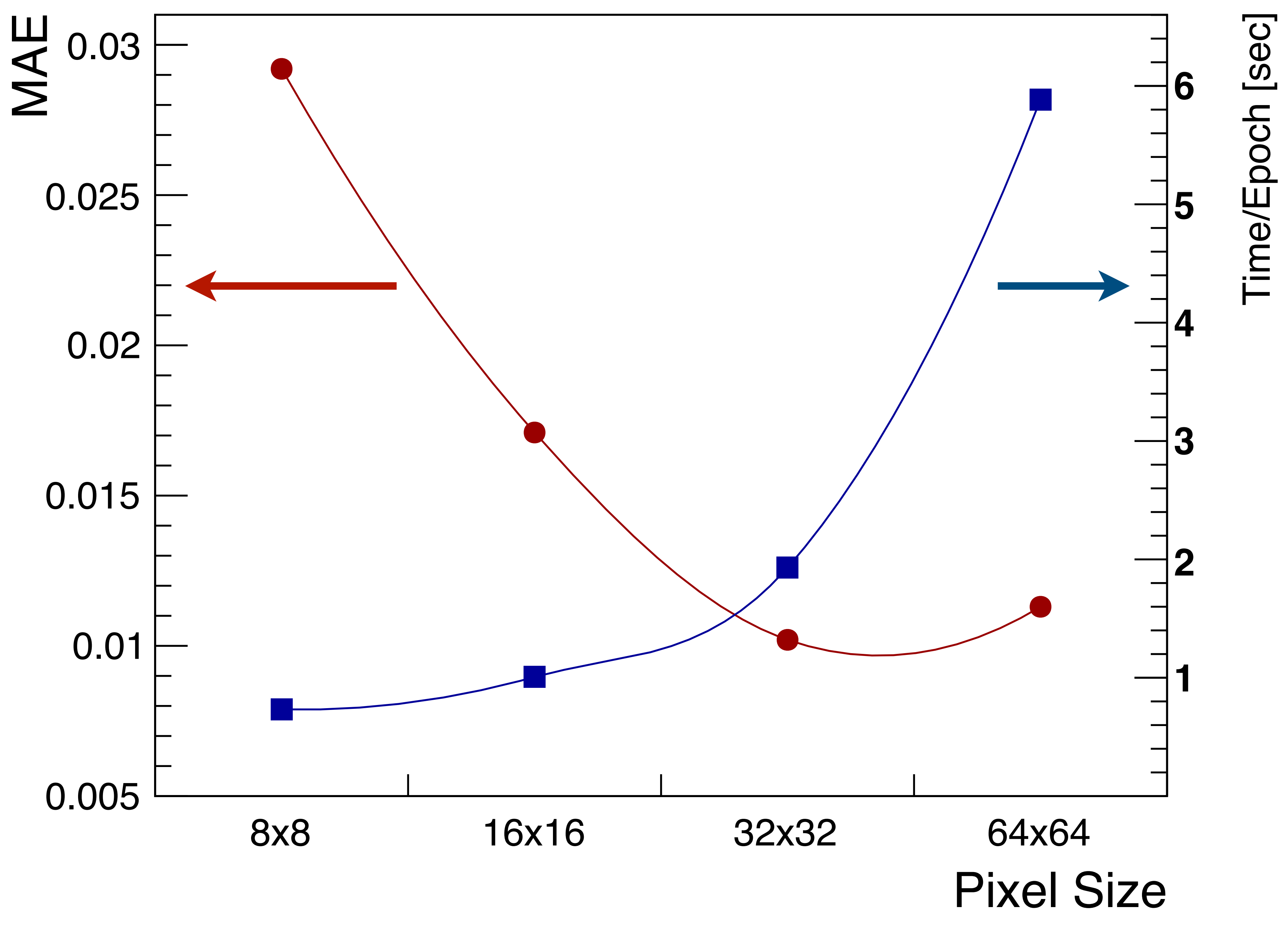


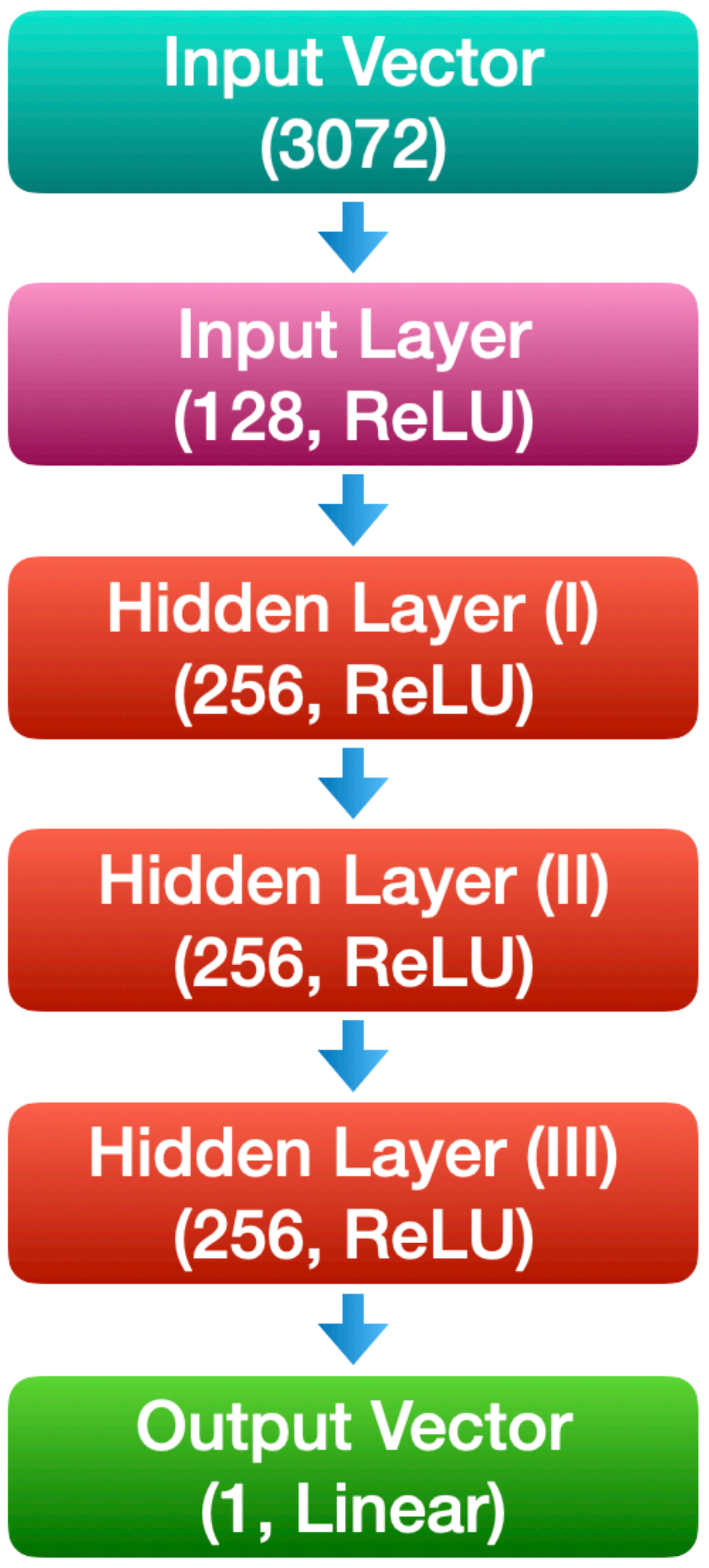
Pb-Pb, $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV, **AMPT Simulation**

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Model

- Each space has 32×32 pixels (grids)
- Total number of pixel points =
  $32 \times 32 \times 3 = 3072$ for each event
- Optimzer: $adam$,
  Loss function: mse
- Max epoch: 100
  Batch Size: 32, callback = [es]
- Training: 2×10^5 Events (~60 GB)
- Validation: $10\,\%$ Events

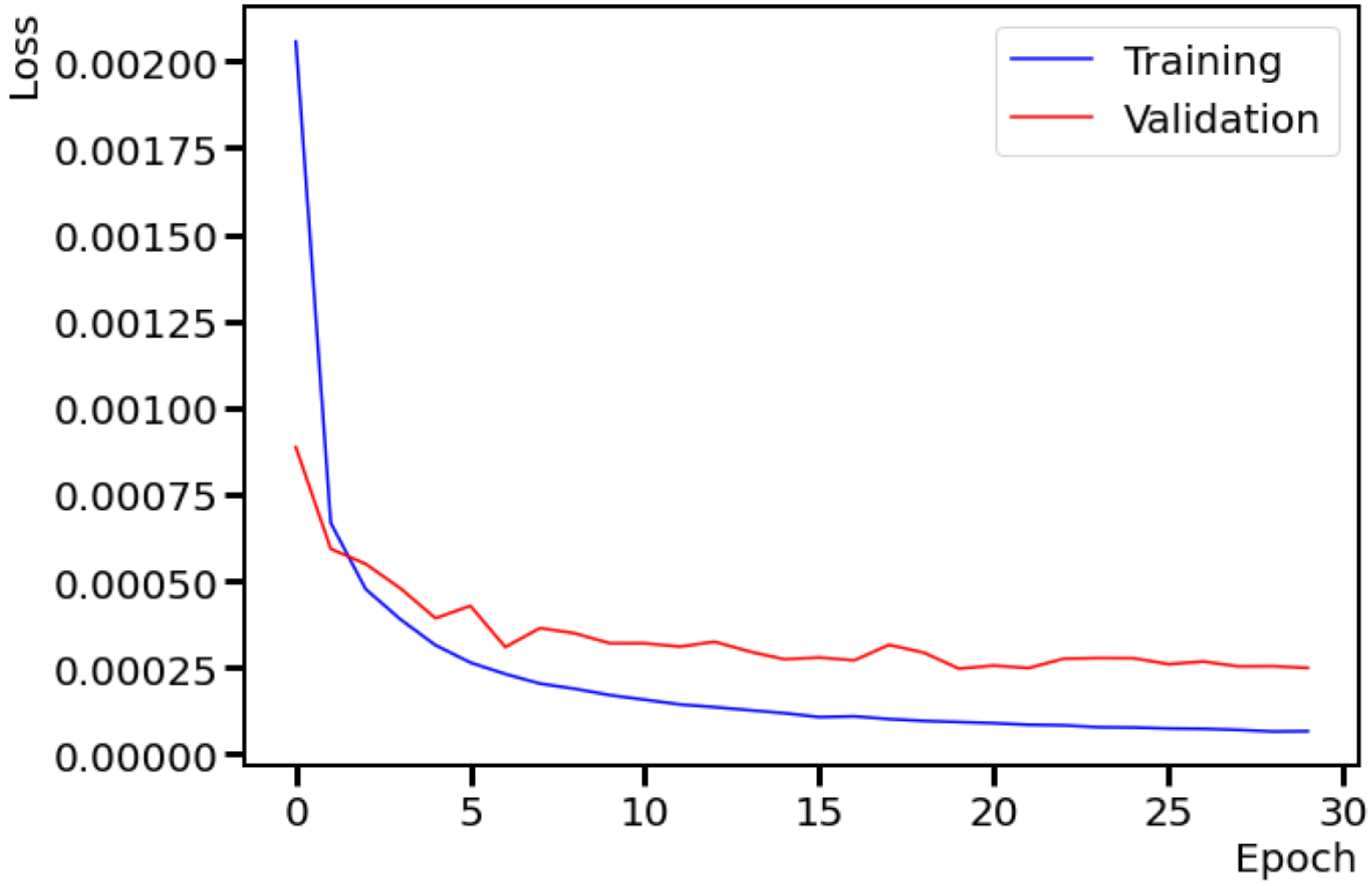


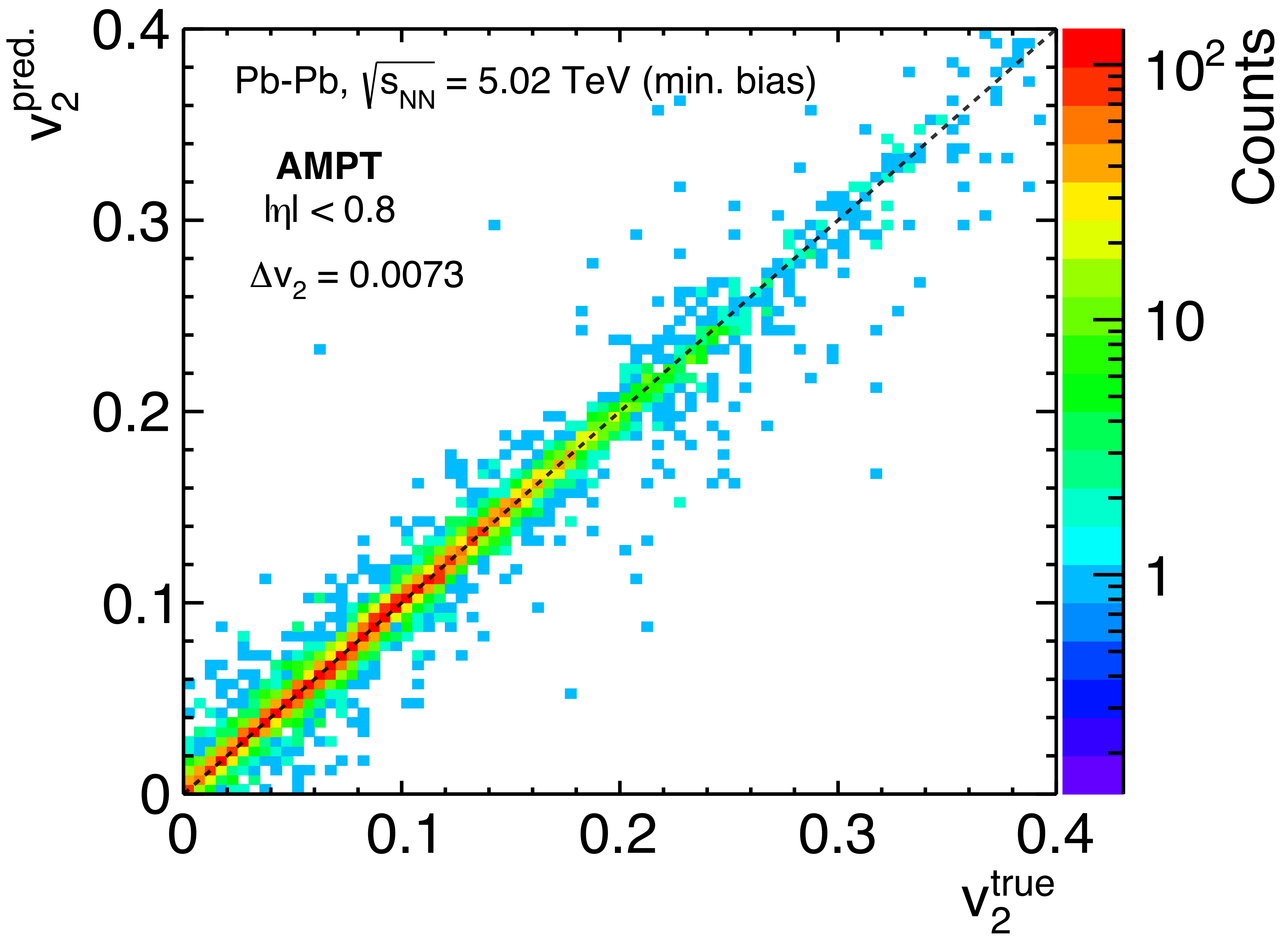


$$\Delta v_2 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} |\, v_{2_n}^{true} - v_{2_n}^{pred.} \,|$$

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Training





$$\Delta v_2 = \frac{1}{N_{\text{events}}} \sum_{n=1}^{N_{\text{events}}} \left| v_{2_n}^{true} - v_{2_n}^{pred.} \right|$$

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, [Phys.Rev.D 105, 114022 (2022)].

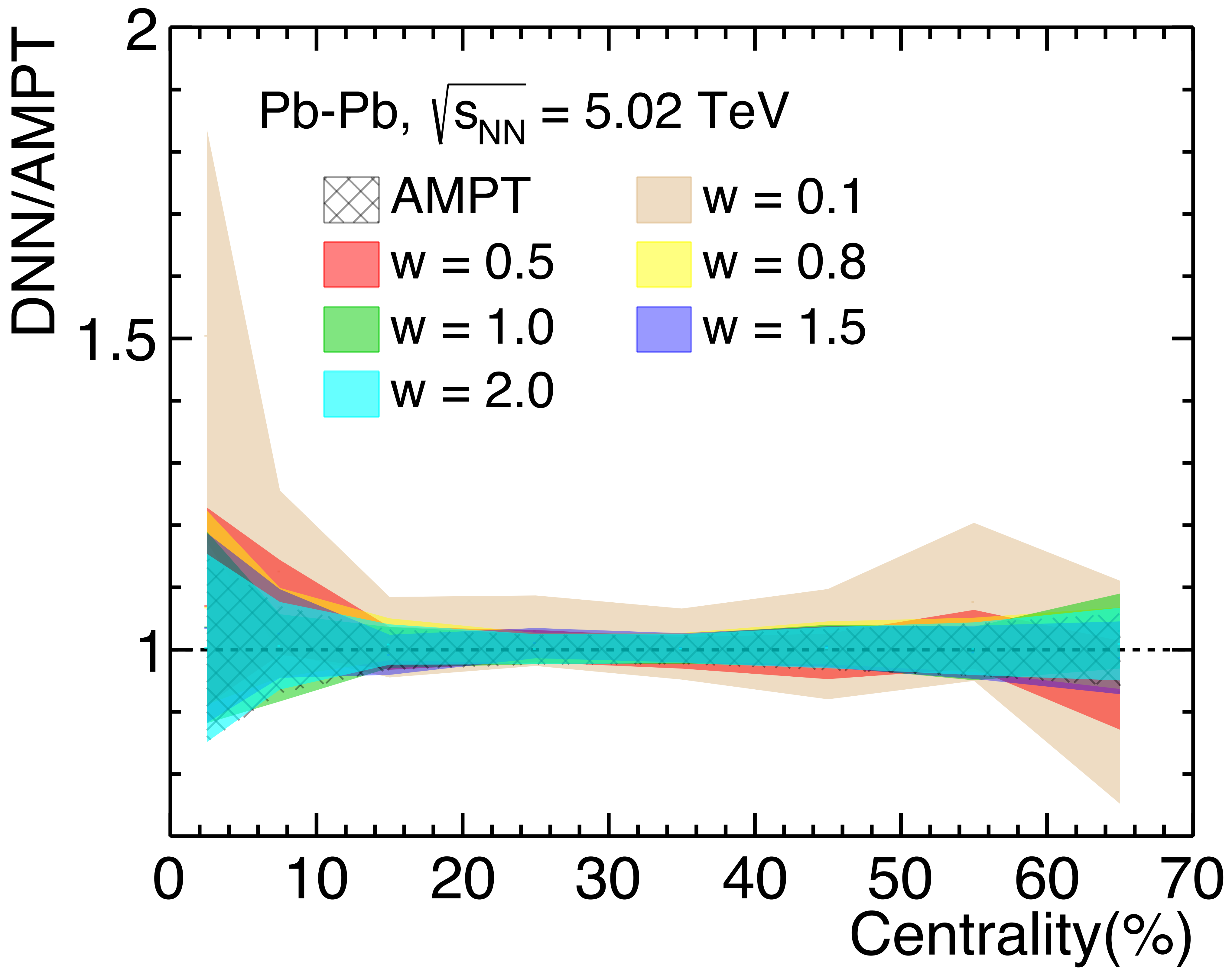
Systematic Uncertainty

- Introduce uncorrelated/random noise to simulation
- For *i*th event, and *j*th feature, the feature value

  $F_{i,j} \leftarrow F_{i,j} + X_{i,j}/w$, where $X_{i,j} \in (-\sigma_j, \sigma_j)$

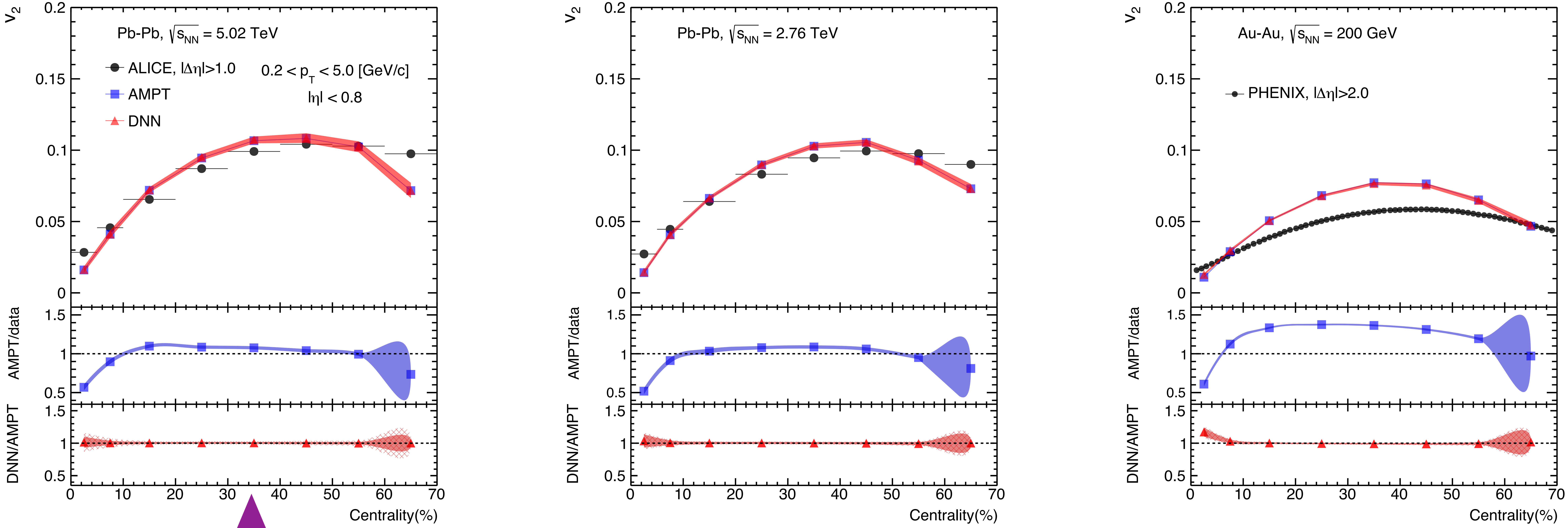
  σ_j = standard deviation, w = noise parameter

- Large $w \rightarrow$ small noise and *vice versa*
- *Stable and accurate prediction → robust model*
- Systematic Uncertainty



N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Centrality dependence



p_{T} dependence

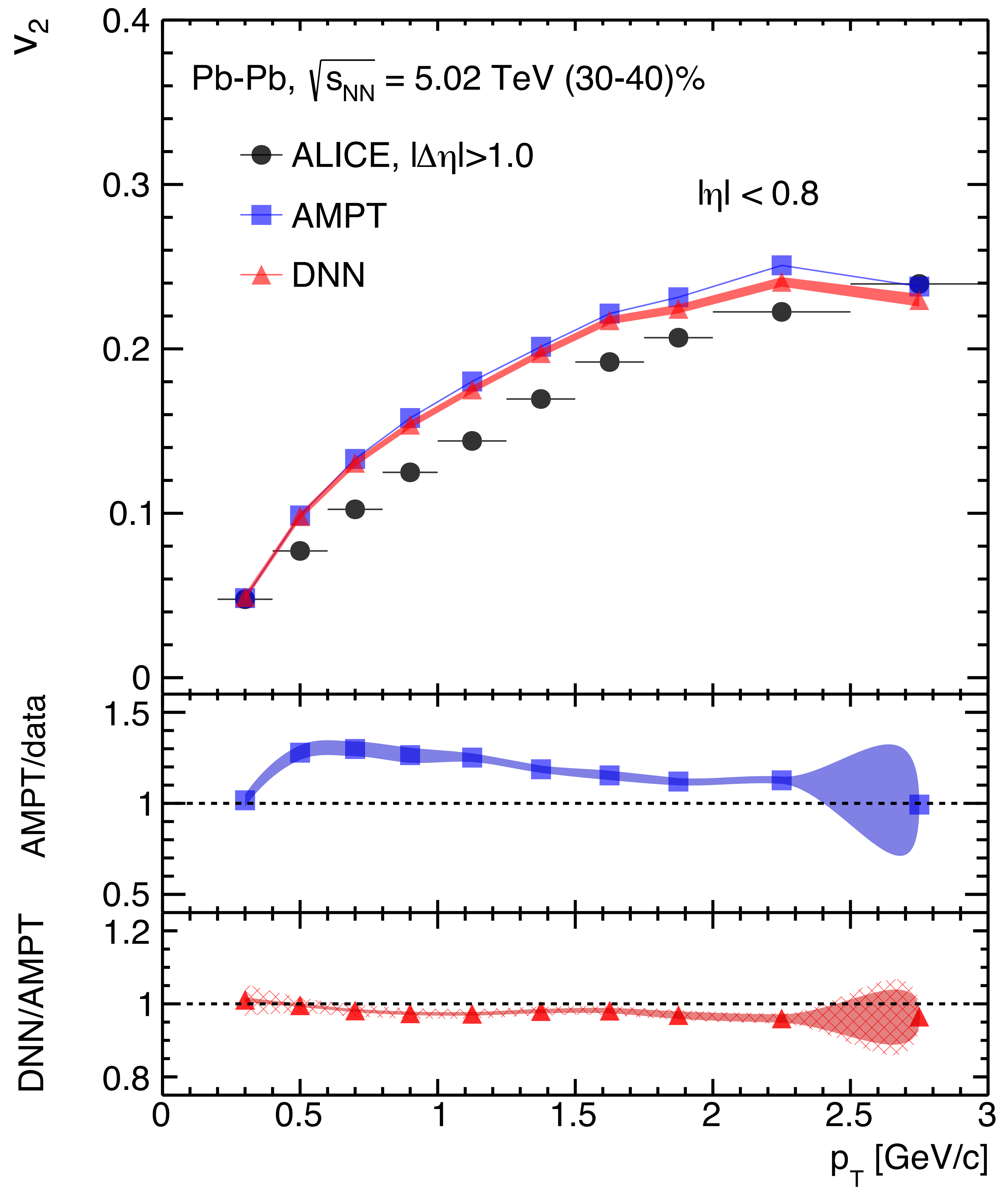
- Good agreement between the simulated and predicted values of v_2
- Trained at LHC energy, applied to RHIC energy
- **DNN preserves centrality, and collision system dependence of v_2**

ALICE, Phys. Rev. Lett. 116, 132302 (2016).

PHENIX, Phys. Rev. C 99, 024903 (2019).

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Transverse momentum dependence



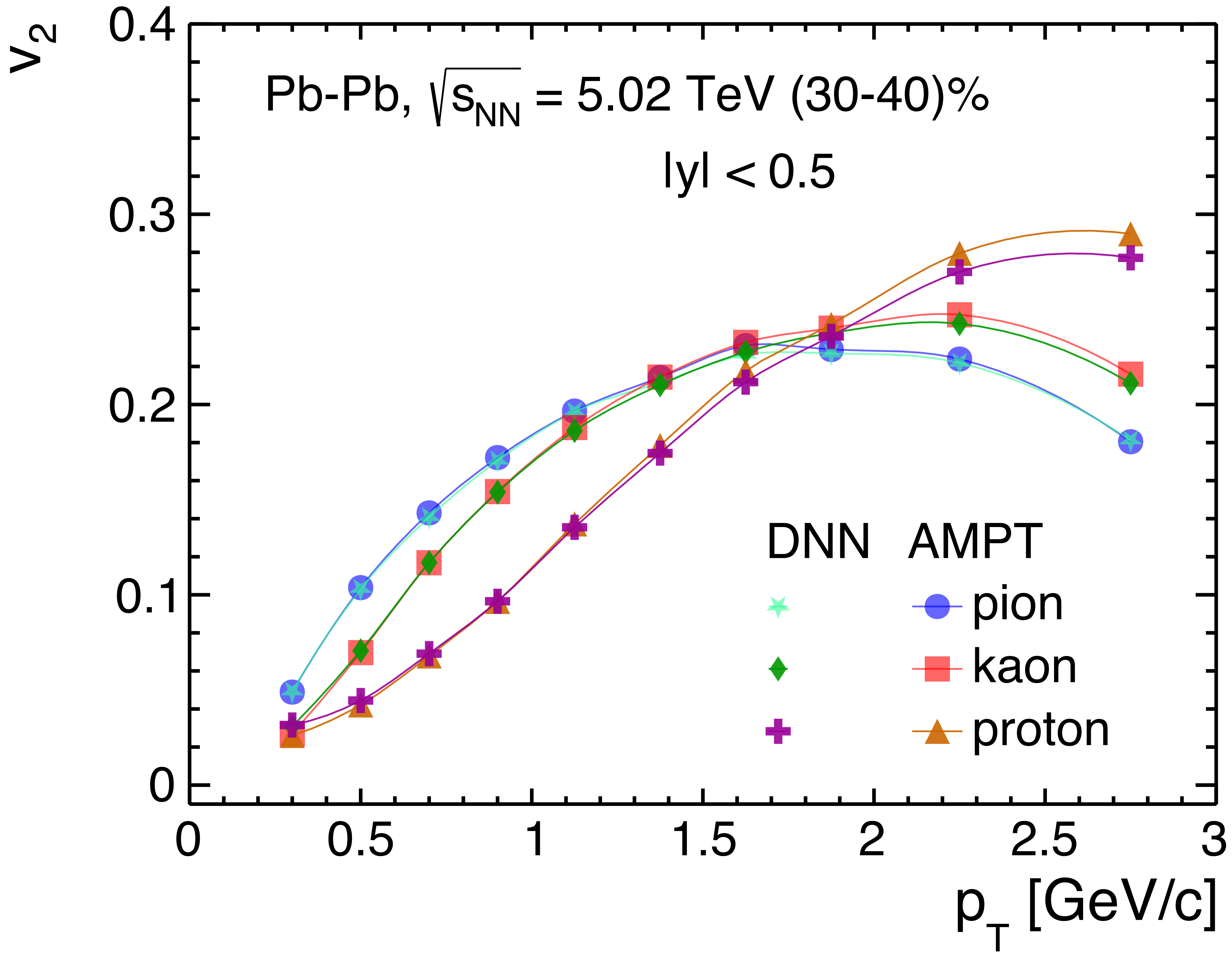
- Training in the range: $0.2 < p_T < 5.0$ [GeV/c]
- Applied to different slices of p_T-bins:

  [0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.5, 1.75, 2.0, 2.5, 3.0]
- **DNN preserves the p_T dependence of v_2**

ALICE, Phys. Rev. Lett. 116, 132302 (2016).

N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, Phys.Rev.D 105, 114022 (2022).

Elliptic flow for identified particles



- Estimation of elliptic flow for pion, kaon, and proton
- DNN is trained with Pb-Pb, $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV (min. bias)
- Good agreement with AMPT results
- **Meson-Baryon level elliptic flow is preserved with DNN**

For future

- Preliminary model based on Monte Carlo simulation entirely
- Propose to test this method with ALICE, Pb-Pb, $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV (min. bias)
- The reaction plane (ψ_R) invariance could be checked
- The effects of detector level (correlated noise) correlations could be checked
- Estimate and remove non-flow effects
- Higher-order flow coefficients
- May lead to explore new ML techniques

Summary

- First Deep Learning based estimator for elliptic flow
- Final state particle kinematic information as input
- DNN preserves p_{T}, centrality, and collision system dependence of v_2
- Excellent prediction accuracy against noisy simulation
- Applicable to both RHIC and LHC energy
- Prediction is much faster and economical

Thank you!